

Bài 1

Bước 1: Gọi hàm `sum_of_numbers(7)`.

Kiểm tra điều kiện cơ sở: $n = 7$, không thỏa mãn điều kiện cơ sở.

Thực hiện câu lệnh trong else: `return 7 + sum_of_numbers(6)`.

Bước 2: Hàm `sum_of_numbers(6)` được gọi.

Kiểm tra điều kiện cơ sở: $n = 6$, không thỏa mãn điều kiện cơ sở.

Thực hiện câu lệnh trong else: `return 6 + sum_of_numbers(5)`.

Bước 3: Hàm `sum_of_numbers(5)` được gọi.

Kiểm tra điều kiện cơ sở: $n = 5$, không thỏa mãn điều kiện cơ sở.

Thực hiện câu lệnh trong else: `return 5 + sum_of_numbers(4)`.

Bước 4: Hàm `sum_of_numbers(4)` được gọi.

Kiểm tra điều kiện cơ sở: $n = 4$, không thỏa mãn điều kiện cơ sở.

Thực hiện câu lệnh trong else: `return 4 + sum_of_numbers(3)`.

Bước 5: Hàm `sum_of_numbers(3)` được gọi.

Kiểm tra điều kiện cơ sở: $n = 3$, không thỏa mãn điều kiện cơ sở.

Thực hiện câu lệnh trong else: `return 3 + sum_of_numbers(2)`.

Bước 6: Hàm `sum_of_numbers(2)` được gọi.

Kiểm tra điều kiện cơ sở: $n = 2$, không thỏa mãn điều kiện cơ sở.

Thực hiện câu lệnh trong else: `return 2 + sum_of_numbers(1)`.

Bước 7: Hàm `sum_of_numbers(1)` được gọi.

Kiểm tra điều kiện cơ sở: $n = 1$, thỏa mãn điều kiện cơ sở.

Trả về 1.

Các hàm được thực hiện theo thứ tự dưới lên (từ bước 7 lên bước 1)

`sum_of_numbers(1) = 1`

`sum_of_numbers(2) = 2 + sum_of_numbers(1) = 2 + 1 = 3`

`sum_of_numbers(3) = 3 + sum_of_numbers(2) = 3 + 3 = 6`

`sum_of_numbers(4) = 4 + sum_of_numbers(3) = 4 + 6 = 10`

`sum_of_numbers(5) = 5 + sum_of_numbers(4) = 5 + 10 = 15`

`sum_of_numbers(6) = 6 + sum_of_numbers(5) = 6 + 15 = 21`

sum_of_numbers(7) = 7 + sum_of_numbers(6) = 7 + 21 = 28

Vậy kết quả cuối cùng là 28.

Bài 2

Bước 1: Gọi hàm fibonacci(8)

Kiểm tra điều kiện cơ sở: $n = 8$, không thỏa mãn điều kiện cơ sở.

Thực hiện câu lệnh trong else: `return fibonacci(7) + fibonacci(6)`.

Bước 2: Hàm fibonacci(7) được gọi.

Kiểm tra điều kiện cơ sở: $n = 7$, không thỏa mãn điều kiện cơ sở.

Thực hiện câu lệnh trong else: `return fibonacci(6) + fibonacci(5)`.

Bước 3: Hàm fibonacci(6) được gọi.

Kiểm tra điều kiện cơ sở: $n = 6$, không thỏa mãn điều kiện cơ sở.

Thực hiện câu lệnh trong else: `return fibonacci(5) + fibonacci(4)`.

Bước 4: Hàm fibonacci(5) được gọi.

Kiểm tra điều kiện cơ sở: $n = 5$, không thỏa mãn điều kiện cơ sở.

Thực hiện câu lệnh trong else: `return fibonacci(4) + fibonacci(3)`.

Bước 5: Hàm fibonacci(4) được gọi.

Kiểm tra điều kiện cơ sở: $n = 4$, không thỏa mãn điều kiện cơ sở.

Thực hiện câu lệnh trong else: `return fibonacci(3) + fibonacci(2)`.

Bước 6: Hàm fibonacci(3) được gọi.

Kiểm tra điều kiện cơ sở: $n = 3$, không thỏa mãn điều kiện cơ sở.

Thực hiện câu lệnh trong else: `return fibonacci(2) + fibonacci(1)`.

Bước 7: Hàm fibonacci(2) được gọi.

Kiểm tra điều kiện cơ sở: $n = 2$, không thỏa mãn điều kiện cơ sở.

Thực hiện câu lệnh trong else: return fibonacci(1) + fibonacci(0).

Bước 8: Hàm fibonacci(1) được gọi.

Kiểm tra điều kiện cơ sở: $n = 1$, thỏa mãn điều kiện cơ sở.

Trả về 1.

Các hàm được thực hiện theo thứ tự dưới lên (từ bước 8 lên bước 1)

fibonacci(1) = 1

fibonacci(2) = fibonacci(1) + fibonacci(0) = 1 + 0 = 1

fibonacci(3) = fibonacci(2) + fibonacci(1) = 1 + 1 = 2

fibonacci(4) = fibonacci(3) + fibonacci(2) = 2 + 1 = 3

fibonacci(5) = fibonacci(4) + fibonacci(3) = 3 + 2 = 5

fibonacci(6) = fibonacci(5) + fibonacci(4) = 5 + 3 = 8

fibonacci(7) = fibonacci(6) + fibonacci(5) = 8 + 5 = 13

fibonacci(8) = fibonacci(7) + fibonacci(6) = 13 + 8 = 21

Vậy kết quả cuối cùng là 21

Bài 3

Bước 1: Gọi hàm power(2, 6)

Kiểm tra điều kiện cơ sở: $n = 6$, không thỏa mãn điều kiện cơ sở.

Thực hiện câu lệnh trong else: return 2 * power(2, 5).

Bước 2: Hàm power(2, 5) được gọi.

Kiểm tra điều kiện cơ sở: $n = 5$, không thỏa mãn điều kiện cơ sở.

Thực hiện câu lệnh trong else: return 2 * power(2, 4).

Bước 3: Hàm power(2, 4) được gọi.

Kiểm tra điều kiện cơ sở: $n = 4$, không thỏa mãn điều kiện cơ sở.

Thực hiện câu lệnh trong else: return 2 * power(2, 3).

Bước 4: Hàm `power(2, 3)` được gọi.

Kiểm tra điều kiện cơ sở: $n = 3$, không thỏa mãn điều kiện cơ sở.

Thực hiện câu lệnh trong `else`: `return 2 * power(2, 2)`.

Bước 5: Hàm `power(2, 2)` được gọi.

Kiểm tra điều kiện cơ sở: $n = 2$, không thỏa mãn điều kiện cơ sở.

Thực hiện câu lệnh trong `else`: `return 2 * power(2, 1)`.

Bước 6: Hàm `power(2, 1)` được gọi.

Kiểm tra điều kiện cơ sở: $n = 1$, không thỏa mãn điều kiện cơ sở.

Thực hiện câu lệnh trong `else`: `return 2 * power(2, 0)`.

Bước 7: Hàm `power(2, 0)` được gọi.

Kiểm tra điều kiện cơ sở: $n = 0$, thỏa mãn điều kiện cơ sở.

Trả về 1.

Các hàm được thực hiện theo thứ tự dưới lên (từ bước 7 lên bước 1)

`power(2, 0) = 1`

`power(2, 1) = 2 * power(2, 0) = 2 * 1 = 2`

`power(2, 2) = 2 * power(2, 1) = 2 * 2 = 4`

`power(2, 3) = 2 * power(2, 2) = 2 * 4 = 8`

`power(2, 4) = 2 * power(2, 3) = 2 * 8 = 16`

`power(2, 5) = 2 * power(2, 4) = 2 * 16 = 32`

`power(2, 6) = 2 * power(2, 5) = 2 * 32 = 64`

Vậy kết quả cuối cùng là 64.

Bài 4

Bước 1: Di chuyển 3 đĩa từ cột A sang cột C, sử dụng cột B như cột trung gian.

Đầu tiên, gọi đệ quy `thap_ha_noi(3, A, B, C)`.

Bước này chia thành 2 phần:

Di chuyển 2 đĩa từ cột A sang cột B, sử dụng cột C như cột trung gian.

Di chuyển đĩa cuối cùng từ cột A sang cột C.

Bước 2: Di chuyển 2 đĩa từ cột A sang cột B, sử dụng cột C như cột trung gian.

Gọi đệ quy `thap_ha_noi(2, A, C, B)`.

Bước này chia thành 2 phần:

Di chuyển 1 đĩa từ cột A sang cột B, sử dụng cột trung gian C.

Di chuyển đĩa cuối cùng từ cột A sang cột B.

Bước 3: Di chuyển 1 đĩa từ cột A sang cột B, sử dụng cột C như cột trung gian.

Với chỉ có 1 đĩa, chỉ cần di chuyển từ A sang B.

In ra thông báo "Chuyển đĩa 1 từ cột A sang cột B".

Bước 4: Di chuyển đĩa cuối cùng từ cột A sang cột B.

In ra thông báo "Chuyển đĩa 2 từ cột A sang cột B".

Bước 5: Di chuyển 1 đĩa từ cột C sang cột B, sử dụng cột A như cột trung gian.

Với chỉ có 1 đĩa, chỉ cần di chuyển từ C sang B.

In ra thông báo "Chuyển đĩa 1 từ cột C sang cột B".

Bước 6: Di chuyển 3 đĩa từ cột A sang cột B, sử dụng cột C như cột trung gian.

Bước này chia thành 2 phần:

Di chuyển 2 đĩa từ cột A sang cột C, sử dụng cột B như cột trung gian.

Di chuyển đĩa cuối cùng từ cột A sang cột B.

Bước 7: Di chuyển 2 đĩa từ cột A sang cột C, sử dụng cột B như cột trung gian.

Gọi đệ quy `thap_ha_noi(2, A, B, C)`.

Tương tự như bước 2, gồm 2 phần:

Di chuyển 1 đĩa từ cột A sang cột C, sử dụng cột trung gian B.

Di chuyển đĩa cuối cùng từ cột A sang cột C.

Bước 8: Di chuyển 1 đĩa từ cột A sang cột C, sử dụng cột B như cột trung gian.

Với chỉ có 1 đĩa, chỉ cần di chuyển từ A sang C.

In ra thông báo "Chuyển đĩa 1 từ cột A sang cột C".

Bước 9: Di chuyển đĩa cuối cùng từ cột A sang cột B.

In ra thông báo "Chuyển đĩa 3 từ cột A sang cột B".

Bước 10: Di chuyển 1 đĩa từ cột C sang cột B, sử dụng cột A như cột trung gian.

In ra thông báo "Chuyển đĩa 1 từ cột C sang cột B".

Kết quả:

Chuyển đĩa 1 từ cột A sang cột C

Chuyển đĩa 2 từ cột A sang cột B

Chuyển đĩa 1 từ cột C sang cột B

Chuyển đĩa 3 từ cột A sang cột C

Chuyển đĩa 1 từ cột B sang cột A

Chuyển đĩa 2 từ cột B sang cột C

Chuyển đĩa 1 từ cột A sang cột C

Chuyển đĩa 4 từ cột A sang cột B

Chuyển đĩa 1 từ cột C sang cột B

Chuyển đĩa 2 từ cột C sang cột A

Chuyển đĩa 1 từ cột B sang cột A

Chuyển đĩa 3 từ cột C sang cột B

Chuyển đĩa 1 từ cột A sang cột C

Chuyển đĩa 2 từ cột A sang cột B

Chuyển đĩa 1 từ cột C sang cột B

Bài 5

Xác định các trường hợp cơ bản:

Nếu không có con vật nào cả (số chó và số gà đều bằng 0), hàm sẽ trả về (0, 0) để thể hiện không có chó và không có gà.

Trong trường hợp số chân là số lẻ, nghĩa là không thể chia đều số chân giữa chó và gà. Do đó, hàm sẽ trả về (-1, -1) để biểu thị rằng bài toán không thể giải quyết được.

Duyệt qua tất cả các trường hợp có thể:

Sử dụng vòng lặp for, hàm duyệt qua từng số lượng chó có thể có, từ 0 đến `tong_so_con` (tổng số con). Đối với mỗi lượt duyệt:

Tính số lượng gà tương ứng bằng cách lấy tổng số con trừ đi số lượng chó.

Kiểm tra nếu tổng số chân của tất cả chó và gà bằng `tong_so_chan`. Nếu đúng, hàm trả về số chó và số gà tìm được.

Nếu không tìm thấy giải pháp:

Trong trường hợp không tìm thấy giải pháp trong vòng lặp, hàm sẽ thử giảm số con và số chân đi 4 (vì mỗi con chó có 4 chân).

Sau đó, hàm gọi đệ quy lại chính nó với các tham số đã giảm.

Nếu từ các bước đệ quy có thể tìm thấy giải pháp, hàm sẽ tăng số chó lên 1 và trả về số chó và số gà.

In kết quả:

Kết quả được in ra, hiển thị số lượng chó và số lượng gà tìm được.